Before beginning the assignment, please carefully read the problem set submission instructions on the course website and create accounts on gradescope and hackerrank.

Note that all exercise numbers from the textbook refer to the **second edition**. If you are using a hard copy of the first edition, you should download the second edition from SpringerLink to look up the exercises.

Written problems

1. The following passage has been encrypted using the Caesar cipher.

```
Nzky kyv yvcg fw kyv arezkfi yv jtivnvu fekf kyv
jzuv fw kyv uvjb r gvetzc jyrigvevi -- kyrk yzxycp
jrkzjwpzex, yzxycp gyzcfjfgyztrc zdgcvdvek kyrk xfvj
kztfeuvifxr-kztfeuvifxr, wvvuzex fe kyv pvccfn wzezjy reu
jnvvk nffu, reu veuj lg ze r bzeu fw jfleucvjjcp jgzeezex
vkyvivrc mfzu rj nv rcc dljk.
```

   Determine the secret key k (the number of places each letter has been advanced in the alphabet). You can solve this by hand, or by writing code to make the task easier. Either way, briefly summarize your strategy, including a description of any code that you write.

2. Throughout this course, we will say that an integer a is an "n-bit (nonnegative) integer" if $0 \leq a < 2^n$ (I will also sometimes use the phrase "exactly n bits long" to mean that $2^{n-1} \leq a < 2^n$, i.e. that a is an n-bit integer but not an $(n-1)$-bit integer).

   The *Data Encryption Standard* (DES) is a private-key encryption algorithm that was a government standard from 1977 to 2002. DES uses 56-bit secret keys. Suppose that Eve attempts a brute-force attack on DES by trying to decrypt an intercepted cipher text with every possible 56-bit key until she finds something that looks like English text. If Eve's system can try 1 billion keys per second, how long would it take her to try all of the keys (and thus be sure to break the encryption)?

   (By 1999, a distributed system was able to break DES encryption in less than 24 hours. DES was replaced in 2002 by a new standard, called AES, which uses keys of at least 128 bits. For "top secret" communication, the government uses AES with 256 bit keys.)

3. Textbook exercise 1.6.

4. Textbook exercise 1.9, parts (a) and (b).

5. Textbook exercise 1.10, parts (a) and (b).

6. Textbook exercise 1.11.

7. This problem is meant to allow you to think about how the sizes of an input to a program influence its runtime, in a concrete setting. The purpose is for you to try to make some educated guesses for now; you do not need to be correct to receive full points.

   (a) Consider the following function. It takes a positive integer N, and returns the number of divisors of N.

```
def divisors(N):
    count = 0
    for d in xrange(1,N+1):
        if N%d == 0:
            count += 1
    return count
```

To be marked as correct on hackerrank, a program must finish within 10 seconds. Try to guess (roughly) how long N can be (in bits) before the function above will not finish in 10 seconds. One useful fact to keep in mind: a modern CPU completes approximately 2^{31} clock cycles per second.

Note. Using `xrange` instead of `range` has essentially the same functionality and prevents certain errors. Search for "xrange versus range" (or similar) to read more about this.

(b) Test your guess by going to the "count divisors" challenge at the following link. The test cases are numbers 0 through 99, where test case k is an integer exactly $k+1$ bits long. Therefore the first test case to fail will tell you the point at which this function no longer can finish in ten seconds.

`https://www.hackerrank.com/m158-2016-demos`

(c) Now consider the following function. Briefly explain why this also correctly calculates the number of divisors of N.

```
def divisors(N):
    count = 0
    d = 1
    while d*d < N:
        if N%d == 0:
            count += 2
        d += 1
    if d*d == N:
        count += 1
    return count
```

(d) Predict how many bits long N must be before the code in part (c) cannot finish in less than 10 seconds. Test your prediction by entering this code and submitting it on hackerrank.

Programming problems

Full specifications and online submission: `http://www.hackerrank.com/m158-2016-pset-1`

8. Write a program which receives an integer N as input, guaranteed to be a product of two 16-bit prime numbers, and determines the two prime factors.

9. Write a program which takes a list of 1024-bit positive integers (i.e. each integer a in the list satisfies $1 \leq a < 2^{1024}$) as input, and prints their greatest common divisor.

   Suggestion. Begin with a function that given the greatest common divisor of any two numbers, then figure out how to use this function to find the GCD of a longer list.

10. (*Extra credit*). Write a program which can solve problem 1 of this assignment automatically (i.e. decrypt a passage of English text encrypted with a Caesar cipher). (Contact me to receive a hint, after you think about what sorts of methods might work).